



# UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE  
United States Patent and Trademark Office  
Address: COMMISSIONER FOR PATENTS  
P.O. Box 1450  
Alexandria, Virginia 22313-1450  
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/482,178	01/12/2000	Theodore M. Osborne, II	1306	6648
22468	7590	06/18/2004	EXAMINER	
CHAPIN & HUANG L.L.C. WESTBOROUGH OFFICE PARK 1700 WEST PARK DRIVE, SUITE 120 WESTBOROUGH, MA 01581			STEELEMAN, MARY J	
ART UNIT		PAPER NUMBER		24
2122				
DATE MAILED: 06/18/2004				

Please find below and/or attached an Office communication concerning this application or proceeding.

<b>Office Action Summary</b>	<b>Application No.</b>	<b>Applicant(s)</b>	
	09/482,178	OSBORNE, II ET AL.	
	<b>Examiner</b>	<b>Art Unit</b>	
	Mary J. Steelman	2122	

**-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --**

### **Period for Reply**

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If the period for reply specified above is less than thirty (30) days, a reply within the statutory minimum of thirty (30) days will be considered timely.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

## Status

1)  Responsive to communication(s) filed on 03/29/2004, 5/27/2004.

2a)  This action is FINAL.                            2b)  This action is non-final.

3)  Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

## **Disposition of Claims**

4)  Claim(s) 1,3-18 and 20-22 is/are pending in the application.  
4a) Of the above claim(s) \_\_\_\_\_ is/are withdrawn from consideration.

5)  Claim(s) \_\_\_\_\_ is/are allowed.

6)  Claim(s) 1,3-18 and 20-22 is/are rejected.

7)  Claim(s) \_\_\_\_\_ is/are objected to.

8)  Claim(s) \_\_\_\_\_ are subject to restriction and/or election requirement.

## Application Papers

9)  The specification is objected to by the Examiner.

10)  The drawing(s) filed on \_\_\_\_\_ is/are: a)  accepted or b)  objected to by the Examiner.

    Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

    Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)  The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

**Priority under 35 U.S.C. § 119**

12)  Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).  
a)  All    b)  Some \* c)  None of:  
1.  Certified copies of the priority documents have been received.  
2.  Certified copies of the priority documents have been received in Application No. \_\_\_\_\_.  
3.  Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

\* See the attached detailed Office action for a list of the certified copies not received

**Attachment(s)**

1)  Notice of References Cited (PTO-892)  
2)  Notice of Draftsperson's Patent Drawing Review (PTO-948)  
3)  Information Disclosure Statement(s) (PTO-1449 or PTO/SB/08)  
Paper No(s)/Mail Date #19, 3/29/04.

4)  Interview Summary (PTO-413)  
Paper No(s)/Mail Date. \_\_\_\_ .  
5)  Notice of Informal Patent Application (PTO-152)  
6)  Other: \_\_\_\_\_

### **DETAILED ACTION**

1. This action is in response to RCE filed 03/29/2004, and Amendment D filed 05/27/204.
2. Per Applicant's request, claim 2 has been canceled. Claim 19 was previously canceled.

Claims 1, 10, 18, and 22 have been amended. Claims 1, 3-18, and 20-22 are pending.

#### ***Information Disclosure Statement***

3. IDS filed 03/29/2004 has been considered.

#### ***Double Patenting***

4. Per Applicant's response in Amendment A, filed 01/21/2003, Applicant agrees to file a terminal disclaimer associated with co-pending application, 09/548203, upon indication of allowance.

#### ***Claim Rejections - 35 USC § 103***

5. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

(a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negated by the manner in which the invention was made.

6. **Claims 1, 3-18 and 20-22** are rejected under 35 U.S.C. 103(a) as being unpatentable over U.S. Patent 5,974,572 to Weinberg et al., in view of U.S. Patent 6,574,578 to Logan.

Weinberg disclosed a Web site (server) analysis program implemented as a collection of software components. Weinberg's invention generates test scenarios (col. 2, lines 38-39 and col. 3, line 7) that emulates multiple concurrent users on the system. Weinberg specified that the content tested is objects (components) (col. 6, lines 8-15) and that a user can retrieve a URL

(content object) from the server (col. 10, lines 7-9.) Weinberg, Col. 33, lines 47-50, "...a code module is provided that automatically generates a load testing scenario..."

Weinberg did not provide extensive details concerning the testing of individual application components. However, Logan did disclose more specific details regarding application components (Col. 2, lines 19-22 and col. 6, lines 52-54) and the utilization of a server to coordinate testing. Logan: Abstract, lines 1-3, "Method and system aspects for utilizing a server to coordinate component testing in an integrated test environment network are described."

**Per claim 1:**

-a) providing test code automatically generated from analysis of the technology based software component that exercises said technology based software component of the application under test;

(Logan, col. 6, line 49-col. 7, line 20, "In accordance with a further embodiment of the present invention, an automatic test case code generator is utilized...When testing components, e.g., JAVABeans...the test case generator generates the skeleton test cases by way of the java.introspection facility (test code automatically generated from analysis of the technology based software component)...to test the API calls within a class and provides an exhaustive core API unit test case of a given class...Thus, a skeleton test suite program is generated that can then be individualized for specific execution framework to automatically execute each test case (test code provided) and post test results to the common repository.")

-b) synchronizing and executing a plurality of instances of the test code, wherein said test code accesses said technology based software component of the application under test over said

computer network and recording performance data on said technology based software component of the application under test; (Logan, col.6, lines 22-23, "...The test suites are run locally ...and the outcomes of the test cases are stored...")

-c) repeating step b) multiple times, with a different number of instances of the test code; (Logan, col.6, line 29, "...repeating an established suite of tests...")

-d) analyzing the recorded performance data to indicate a performance characteristic of said technology based software component of the application under test in response to load. (Logan, col. 7, line 47-49, "...generates the test results transaction...posted to the database server...")

Therefore, it would have been obvious, to one of ordinary skill in the art, at the time of the invention, to modify Weinberg's invention which generates a load test on a server by including Logan's invention which disclosed more details regarding the testing of application components (col. 2, lines 39-47) in a distributed environment (fig. 1), because extensive testing of components is useful in comparing expected outcome versus actual outcome, interactive tests can provide feedback, and checklist testing can ensure quality checks are performed in a complete and consistent manner (Logan, col.7, lines 25-40).

**Per claim 3:**

-application under test is an object oriented language and the step of providing test code comprises providing test code to exercise one software component of the application.  
(Weinberg, col. 11, lines 61-67, col. 19, lines 60-64, and col. 20, lines 4-7. )

**Per claim 4:**

-the step of synchronizing comprises starting each instance of the test code at the same time.  
(Weinberg, Col. 20, lines 22-26, “A task manager process...handles issues related to the management of the threads, including the synchronization...”).

**Per claim 5:**

-the step of synchronizing and executing comprises executing a portion of the plurality of instances of test code on a first computer and a portion of the plurality of instance of test code on a second computer connected to the network. (Weinberg, Fig. 25 and col. 32, lines 51-58, “...multiple Vusers (i.e., multiple instances of the Vuser executable) can be run simultaneously on a single workstation...This produces a load in which multiple client requests can...be pending at-a-time...”).

**Per claim 6:**

Preparing a graphical display having as an independent variable the number of instances of the test code and the dependent variable is the performance data. (Weinberg, Fig. 26 and col. 18, lines 64-67, “...the Action Tracker plug-in communicates with the Web sites...to retrieve server access log files for performing Web site activity analyses. Also, col. 33, lines 3-8, “...user is

presented with a set of graphical reports that allow the user (to) evaluate the site's performance...”)

**Per claim 7:**

Preparing a graphical display having as an independent variable the number of instances of the test code and the dependent variable is derived from the performance data. (Weinberg, Fig. 26, col. 32, line 64 – col. 33, line 8, “...records various performance-related characteristics of these responses. These characteristics include...response times to individual client requests...the user is presented with a set of graphical reports...the user can...compare response times of different site components...”)

**Per claim 8:**

Weinberg disclosed a Web site (server) analysis program implemented as a collection of software components. Weinberg's invention generates test scenarios (col. 2, lines 38-39 and col. 3, line 7) that emulates multiple concurrent users on the system. Weinberg specified that the content tested is objects (components) (col. 6, lines 8-15) and that a user can retrieve a URL (content object) from the server (col. 10, lines 7-9.)

Weinberg did not provide extensive details concerning the testing of individual application components. However, Logan did disclose:

-application under test is resident on a first server within the network and the application has a remote interface and the test code is resident on at least a second computer within the network and exercises the software component of the application under test using the remote interface of

the application under test. (Logan, col. 7, lines 41-49, "...client instantiates the software objects being tested...passes to ...server..." Also see fig. 1.)

Therefore, it would have been obvious, to one of ordinary skill in the art, at the time of the invention, to modify Weinberg's invention which generates a load test on a server by including Logan's invention which disclosed more details regarding the testing of application components (col. 2, lines 39-47) in a distributed environment (fig. 1), because extensive testing of components is useful in comparing expected outcome versus actual outcome, interactive tests can provide feedback, and checklist testing can ensure quality checks are performed in a complete and consistent manner (Logan, col.7, lines 25-40).

**Per claim 9:**

-The step of analyzing includes displaying the analyzed data to a human user using a graphical user interface. (Weinberg, Col. 33, lines 3-4, "the user is presented with a set of graphical reports...")

**Per claim 10:**

Weinberg disclosed a Web site (server) analysis program implemented as a collection of software components. Weinberg's invention generates test scenarios (col. 2, lines 38-39 and col. 3, line 7) that emulates multiple concurrent users on the system. Weinberg specified that the content tested is objects (components) (col. 6, lines 8-15) and that a user can retrieve a URL (content object) from the server (col. 10, lines 7-9.)

Weinberg did not provide extensive details concerning the testing of individual application components. However, Logan did disclose:

-a) specifying test conditions through a user interface to a test system;

(Logan: col. 4, lines 51-52, The ITE client then issues test suite staging requests to the ITE server.”)

-b) initiating, across a network, through a user interface to the test system the gathering of test data on the performance of at least one technology based software component of the application under test at a plurality of load conditions, the load conditions including running test code automatically generated from analysis of the technology based software component;

(Logan, col. 6, line 49-col. 7, line 20, “In accordance with a further embodiment of the present invention, an automatic test case code generator is utilized...When testing components, e.g., JAVABeans...the test case generator generates the skeleton test cases by way of the java.introspection facility (test code automatically generated from analysis of the technology based software component)...to test the API calls within a class and provides an exhaustive core API unit test case of a given class...Thus, a skeleton test suite program is generated that can then be individualized for specific execution framework to automatically execute each test case (test code provided) and post test results to the common repository.” See Logan, figs. 7 & 11, networked testing. Col. 7, lines 16-20, “Thus, a skeleton test suite program is generated...to automatically execute each test case and post test results (gathering of test data) to the common repository.” Col. 7, lines 43-49, “The framework on the ITE client instantiates...objects being

tested, drives the execution of test suites, captures responses...generates the test results transaction, which it then passes to the ITE application server to be posted to the database server.” (networked transactions))

-c) specifying through a user interface to the test system the output format of the test data; (Logan: col. 2, lines 38-47, “...integrates test environment...Client **user interface**, a data base server, application server, web server and client application code integrates the tools and data include facilities for: developing and executing suites of related test cases, developing and executing testing checklists...and other documentation defining the testing standards (specify output format), methods and procedures and centralized reporting facilities.” (emphasis added))

-d) displaying in the specified format the response of said at least one technology based software component of the application under test to load.

(Logan, col. 7, lines 41-60, and fig. 6, “...AdriverPanel constructs a TestFrame frame window that contains a TestingPanel object which, in turn, contains any graphical elements of the component software being tested...” And Col. 6, lines 52-56, “JAVABeans” (technology based software component).)

Therefore, it would have been obvious, to one of ordinary skill in the art, at the time of the invention, to modify Weinberg’s invention which generates a load test on a server by including Logan’s invention which disclosed more details regarding the testing of application components (col. 2, lines 39-47) in a distributed environment (fig. 1), because extensive testing of components is useful in comparing expected outcome versus actual outcome, interactive tests

can provide feedback, and checklist testing can ensure quality checks are performed in a complete and consistent manner (Logan, col.7, lines 25-40).

**Per claim 11:**

-the specified format is a graphical format indicating response time as a function of load conditions. (Weinberg, Col. 33, line 3-4, "...the user is presented with a set of graphical reports...")

**Per claim 12:**

-the specified graphical format is a Hi-Lo plot. (Weinberg, Col. 33, line 3-4, "...the user is presented with a set of graphical reports...").

**Per claim 13:**

-the step of gathering data under a plurality of load conditions comprises initiating the execution of a plurality of copies of a test program, with the number of copies executing simultaneously relates to the load condition. (Weinberg, Col. 33, lines 24-39, "To define a scenario, the user initially uses the Web Vuser Generator component to generate the Web scripts to be included within the scenario...the user specifies such details as the number of Vusers, the Web script to be run...and the number of consecutive times...The user can also define one or more Sgroups, and can specify various testing parameters...").

**Per claim 14:**

-the step of specifying an output format includes specifying a method by which response is measured. (Weinberg, Col. 32. line 64 – col. 33, line 8, “Vuser monitors the Web site’s responses to the client requests submitted by that Vuser, and records various performance-related characteristics of these responses...These characteristics include, for example, response times...user is presented with a set of graphical reports that allow the user (to) evaluate the site’s performance...”.)

**Per claim 15:**

-the step of gathering test data includes recording the execution time between selected points in the test program for each simultaneously executing copy of the test program and analyzing the recorded execution times for all copies of the test program. (Weinberg, Col. 33, line 5-6, “...user can...compare response times of different site components...”.)

**Per claim 16:**

-the step of analyzing comprises determining the average and maximum execution times for each of the load conditions. (Weinberg, Col. 32. line 64 – col. 33, line 8, “Vuser monitors the Web site’s responses to the client requests submitted by that Vuser, and records various performance-related characteristics of these responses...These characteristics include, for example, response times...user is presented with a set of graphical reports that allow the user (to) evaluate the site’s performance...”)

**Per claim 17:**

-the computerized application under test comprises software resident on a server controlling access to a computerized database; (Weinberg, Col. 3, lines 5-15, "...based on information stored within a server access log file. The server access log file is ...generated by...Web server. These log files contain information about accesses..." Also see FIG. 13, item 180 and col. 25, lines 3-11, "...specifies ...other Web server extension component 180 to which the form is addressed...Astra extract the dataset...")

-the server is connected to a network and the application under test is simultaneously accessed by a plurality of clients over the network; (Weinberg, FIG. 11 and FIG. 25)

-the test system is resident on at least a second server connected to the network. (Weinberg, FIGs. 11 & 25)

**Per claim 18:**

Weinberg disclosed a Web site (server) analysis program implemented as a collection of software components. Weinberg's invention generates test scenarios (col. 2, lines 38-39 and col. 3, line 7) that emulates multiple concurrent users on the system. Weinberg specified that the content tested is objects (components) (col. 6, lines 8-15) and that a user can retrieve a URL (content object) from the server (col. 10, lines 7-9.)

Weinberg did not provide extensive details concerning the testing of individual application components. However, Logan did disclose:

-a) providing test code automatically generated from analysis of the technology based software component to exercise a selected technology based software component:

(Logan, col. 6, line 49-col. 7, line 20, "In accordance with a further embodiment of the present invention, an automatic test case code generator is utilized... When testing components, e.g., JAVABeans... the test case generator generates the skeleton test cases by way of the java.introspection facility (test code automatically generated from analysis of the technology based software component)... to test the API calls within a class and provides an exhaustive core API unit test case of a given class... Thus, a skeleton test suite program is generated that can then be individualized for specific execution framework to automatically execute each test case (test code provided) and post test results to the common repository.")

-b) creating a first plurality of copies of the test code; (Logan, col. 6, line 67 – col. 7, line 1, "... test case code generator generates the skeleton test cases by way of the java.introspection facility...")

-c) simultaneously executing the first plurality of copies of test code while recording times between events in each of the first plurality of copies of test code, wherein said test code accesses said technology based software component over the computer network; (Logan, col. 7, lines 30-35. See "background automated tests." FIGs. 1 & 3 show networked accesses.)

-d) creating a second plurality of copies of test code; (Logan, col. 7, lines 16-20, "... a skeleton test suite program is generated that can then be individualized for specific test cases of the

component and integrated into the testing execution framework to automatically execute each test case and post test results...”)

-e) simultaneously executing the second plurality of copies of test code while recording times between events in each of the second plurality of copies of test code; (Logan, col. 2, lines 42-47, “Testing tools include facilities for: developing and executing suites of related test cases...”)

-f) repeating a predetermined number of times the steps of creating plural copies of the test code and simultaneously executing the plural copies while recording event times; (Logan, col. 4, lines 57-60, and col. 6, line 29, “...repeating an established suite of tests...”)

-g) analyzing the recorded times to present information on the performance of the technology based software component of the application under test as a function of load. (Logan, col. 7, lines 33-34, “...comparison between the expected outcome and the actual outcome can be performed...”)

Therefore, it would have been obvious, to one of ordinary skill in the art, at the time of the invention, to modify Weinberg’s invention which generates a load test on a server by including Logan’s invention which disclosed more details regarding the testing of application components (col. 2, lines 39-47) in a distributed environment (FIG. 1), because extensive testing of components is useful in comparing expected outcome versus actual outcome, interactive tests can provide feedback, and checklist testing can ensure quality checks are performed in a complete and consistent manner (Logan, col.7, lines 25-40).

**Per claim 20:**

Weinberg disclosed a Web site (server) analysis program implemented as a collection of software components. Weinberg's invention generates test scenarios (col. 2, lines 38-39 and col. 3, line 7) that emulates multiple concurrent users on the system. Weinberg specified that the content tested is objects (components) (col. 6, lines 8-15) and that a user can retrieve a URL (content object) from the server (col. 10, lines 7-9).

Weinberg did not provide extensive details concerning the testing of individual application components. However, Logan did disclose:

-software component has a plurality of functions therein and the test code exercises functions of the software components. (Logan, col. 6, lines 61-63, "...generate a basic core API test skeleton program that systematically exposes all protected methods...")

Therefore, it would have been obvious, to one of ordinary skill in the art, at the time of the invention, to modify Weinberg's invention which generates a load test on a server by including Logan's invention which disclosed more details regarding the testing of application components (col. 2, lines 39-47) in a distributed environment (FIG. 1), because extensive testing of components is useful in comparing expected outcome versus actual outcome, interactive tests can provide feedback, and checklist testing can ensure quality checks are performed in a complete and consistent manner (Logan, col.7, lines 25-40).

**Per claim 21:**

Weinberg disclosed a Web site (server) analysis program implemented as a collection of software components. Weinberg's invention generates test scenarios (col. 2, lines 38-39 and col. 3, line 7) that emulates multiple concurrent users on the system. Weinberg specified that the content tested is objects (components) (col. 6, lines 8-15) and that a user can retrieve a URL (content object) from the server (col. 10, lines 7-9).

Weinberg did not provide extensive details concerning the testing of individual application components. However, Logan did disclose:  
-the events at which times are recorded includes times at which commands are issued to access functions of the software components and times at which execution of the commands are completed. (Logan, col. 7, lines 25-40.)

Therefore, it would have been obvious, to one of ordinary skill in the art, at the time of the invention, to modify Weinberg's invention which generates a load test on a server by including Logan's invention which disclosed more details regarding the testing of application components (col. 2, lines 39-47) in a distributed environment (FIG. 1), because extensive testing of components is useful in comparing expected outcome versus actual outcome, interactive tests can provide feedback, and checklist testing can ensure quality checks are performed in a complete and consistent manner (Logan, col.7, lines 25-40).

**Per claim 22:**

- a) coordination software; (Weinberg, Abstract lines 1-4 and Logan, col. 2, lines 19-22.)

-b) at least one code generator, receiving as an input commands from the coordination software and having as an output client test code automatically generated from analysis of the technology based software component;

(Weinberg, Abstract, lines 20-22, "A Load Wizard module uses this activity data to generate testing scenarios...")

-c) at least one test engine, receiving as an input commands from the coordination software, the test engine comprising a computer server having a plurality of threads thereon, each thread executing an instance of the client test code, wherein said client test code accesses said technology based software component over a computer network; (Weinberg, FIGs. 25&30. See FIGs. 7 & 11 for examples of network accesses.)

-d) at least one data log having computerized memory, the memory holding timing data created by the instances of the client test code in the plurality of threads; (Weinberg, col. 32, lines 64-67.)

-e) at least one data analyzer software, operatively connected to the data log, having an output that represents performance of the technology based software component of the application under test in response to load. (Weinberg, col. 33, lines 3-4, "...user is presented with a set of graphical reports...")

Weinberg disclosed a Web site (server) analysis program implemented as a collection of software components. Weinberg's invention generates test scenarios (col. 2, lines 38-39 and col.

3, line 7) that emulates multiple concurrent users on the system. Weinberg specified that the content tested is objects (components) (col. 6, lines 8-15) and that a user can retrieve a URL (content object) from the server (col. 10, lines 7-9.) Weinberg, Col. 33, lines 47-50, "...a code module is provided that automatically generates a load testing scenario..."

Weinberg did not provide extensive details concerning the testing of individual application components. However, Logan did disclose more specific details regarding application components (Col. 2, lines 19-22 and col. 6, lines 52-54) and the utilization of a server to coordinate testing.

Logan: col. 6, line 49-col. 7, line 20, "In accordance with a further embodiment of the present invention, an automatic test case code generator is utilized... When testing components, e.g., JAVABeans...the test case generator generates the skeleton test cases by way of the java.introspection facility (test code automatically generated from analysis of the technology based software component)...to test the API calls within a class and provides an exhaustive core API unit test case of a given class... Thus, a skeleton test suite program is generated that can then be individualized for specific execution framework to automatically execute each test case (test code provided) and post test results to the common repository."

Therefore, it would have been obvious, to one of ordinary skill in the art, at the time of the invention, to modify Weinberg's invention which generates a load test on a server by including Logan's invention which disclosed more details regarding the testing of application components (col. 2, lines 39-47) in a distributed environment (FIG. 1), because extensive testing of components is useful in comparing expected outcome versus actual outcome, interactive tests

can provide feedback, and checklist testing can ensure quality checks are performed in a complete and consistent manner (Logan, col. 7, lines 25-40).

***Response to Arguments***

7. Applicant's amendments and arguments filed 27 May 2004 (paper #21) have been fully considered but they are not persuasive.

**(A) Applicants have argued, in substance, the following:**

As Applicant has noted on page 7, 3<sup>rd</sup> paragraph, of Amendment D, "Neither Logan nor Weinberg, taken alone or in combination, disclose or suggest the use of test code automatically generated from analysis of the technology based software component in order to perform testing of the technology based software component.", limitations newly added to all independent claims.

**Examiner's Response:**

Examiner disagrees. Logan does disclose "the use of test code automatically generated from analysis of the technology based software component in order to perform testing of the technology based software component." As noted in claim 1 above, Logan' technique uses the java.introspection facility. Logan: col. 7, lines 1-20), "The test case code generator portion of the ITE client suitably performs automatic test case generation... Thus a skeleton test suite program is generated that can then be individualized for specific test cases of the component..."

Weinberg does disclose testing "technology based software components" over a "network". Weinberg's invention disclosed generating test scenarios (col. 3, lines 6-7).

Weinberg disclosed content as (col. 6, lines 11-13) "content objects include HTML documents, GIF files, sound files, video files, JAVA applets and aglets, and downloadable applications...(technology based software components)."

Logan disclosed (Abstract, lines 1-2) "utilizing a server to coordinate component testing in an integrated test environment network..." Logan disclosed (col. 6, lines 52-56, "When testing components, e.g., JAVABeans, each executable class within the component must be tested."

Examiner maintains the rejections of claims 1, 3-18, and 20-22.

***Conclusion***

8. The prior art made of record and not relied upon is considered pertinent to applicant's disclosure.

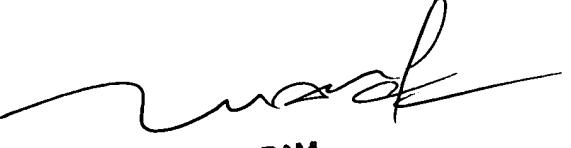
Additionally note, IDS filed 31 March 2004: US Patent 6473794 B1 to Guheen et al. Col. 12, lines 45-65, cols. 60, 66, 67, 127, 130, 131, 175, 271, 272 regarding electronic testing and web site testing tools.

9. Any inquiry concerning this communication or earlier communications from the examiner should be directed to Mary Steelman, whose telephone number is (703) 305-4564. The examiner can normally be reached Monday through Thursday, from 7:00 A.M. to 5:30 P.M. If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Tuan Dam can be reached on (703) 305-4552.

The fax phone number is (703) 872-9306 for regular communications and for After Final communications. Any inquiry of a general nature or relating to the status of this application or proceeding should be directed to the receptionist whose telephone number is (703) 305-3900.

Mary Steelman

06/07/2004



TUAN DAM  
SUPERVISORY PATENT EXAMINER